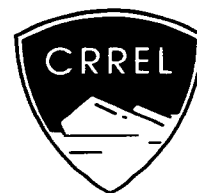


91-15

SPECIAL REPORT

AD-A244 099



2

# User's Manual for ESTK1D.FOR and ESTK2D.FOR Wavenumber Estimation Routines

Mark L. Moran

October 1991

DTIC  
ELECTE  
JAN 8 1992  
S B D

DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

*For conversion of SI metric units to U.S./British customary units of measurement consult ASTM Standard E380, Metric Practice Guide, published by the American Society for Testing and Materials, 1916 Race St., Philadelphia, Pa. 19103.*

*This report is printed on paper that contains a minimum of 50% recycled material.*

# Special Report 91-15



**U.S. Army Corps  
of Engineers**  
Cold Regions Research &  
Engineering Laboratory

## **User's Manual for ESTK1D.FOR and ESTK2D.FOR Wavenumber Estimation Routines**

Mark L. Moran

October 1991

**92-00121**



Prepared for  
OFFICE OF THE CHIEF OF ENGINEERS

Approved for public release; distribution is unlimited.

92 1 2 1 7

## PREFACE

This report was prepared by Mark L. Moran, Geophysicist, Geophysical Sciences Branch, Research Division, U.S. Army Cold Regions Research and Engineering Laboratory. Funding was provided by the Office of the Chief of Engineers under DA Project 4A762784AT42, *Cold Regions Engineering Technology*; Task FS; Work Unit 017, *Algorithms for Seismic/Acoustic Sensors*.

These programs and manual are subject to revision and further development. A copy of the most recent source code or a copy of the executable programs for MS DOS 80386 computers is available from the author, USA CRREL-RG, 72 Lyme Road, Hanover, N.H. 03755-1290, 603-646-4274; mmoran@hanover-crrel.army.mil

Questions and comments concerning this manual, program performance or operation are encouraged as well.

The author reserves the right to revise this software and manual without notification or obligation to any user. All software described here is designed and intended for research purposes only. *No warranty is expressed or implied.*



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution//	
Availability Codes	
Dist.	Avail and/or Special
A-1	

## CONTENTS

1. Introduction .....	1
2. Hardware and software requirements .....	1
2.1. Hardware requirements for use of executable files .....	1
2.2. Software requirements for use of executable files .....	2
2.3. Hardware requirements for creation of executable program based on generic source code .....	2
2.4. Software requirements for creation of executable program based on generic source code .....	2
3. Program description .....	2
3.1. Description of ESTK1D.FOR .....	3
3.2. Description of ESTK2D.FOR .....	4
4. Command file options and structure of input files .....	4
4.1. Command file .....	5
4.2. Data file .....	8
5. Operation of ESTK1D and ESTK2D .....	8
5.1. Installation .....	8
5.2. Invocation methods .....	8
5.3. Program status reports and required terminal input .....	9
5.4. Possible error conditions .....	9
6. Description and explanation of output file format .....	10
7. Program assumptions and performance .....	10
8. Literature cited .....	11
Appendix A: Examples of program output .....	13
Appendix B: List of required files .....	15
Abstract .....	17

## ILLUSTRATIONS

### Figure

1. Flow diagram of ESTK1D .....	3
2. Flow diagram of ESTK2D .....	4
3. Example command file needed to operate ESTK1D .....	5
4. Example command file needed to operate ESTK2D .....	7

# **Users' Manual for ESTK1D.FOR and ESTK2D.FOR**

## **Wavenumber Estimation Routines**

MARK L. MORAN

### **1. INTRODUCTION**

The programs described in this manual produce estimates of the wavenumber of a spatially sampled wavefield obtained from an array of sensors. These estimates are carried out in the frequency-wavenumber (F-K) domain utilizing widely accepted beamforming procedures. Such multidimensional spectral estimation algorithms are generally not available from commercial software vendors. Historically, there is a variety of problems to which these types of F-K estimation programs have been applied. In seismology, F-K estimation procedures utilizing spatially sampled wavefield data sets from an array of geophones are frequently used to determine the epicenter of regional seismic events (Capon et al. 1967). With sonar, passive arrays of hydrophones locate and track submarines and surface vessels by spatially sampling their acoustic wavefields and applying F-K analysis processing (Hahn 1975). Other applications range from tracking aircraft with radar to mapping stellar sources of radio wave emissions. In short, F-K estimation analysis is applicable to any problem in which the unknown wavenumbers in a propagating wave field are needed and there is only limited information on the spatial characteristics of the field. Theoretical details and a general discussion of wavenumber estimation based on F-K algorithms may be found in Capon (1969) or Johnson (1982).

The beamforming procedures used in the two programs described in this manual may be thought of as the spatial equivalent of a Fourier transform based on an uneven, sparsely sampled data set. Reliable quantitative estimates of the bias and variance in spatial spectral estimates of this type are not practically implementable without a detailed prior knowledge of the wavefield's spatial distribution (but if one already has such information then it obviates the need for estimation!). Thus, the resolution, bias and variance in the wavenumber esti-

mates should cause the user to view the results with a cautious and conservative consideration.

This manual details the structure and use of the Fortran programs ESTK1D and ESTK2D. It is not intended to be a comprehensive discussion of array design or array signal processing. ESTK1D and ESTK2D implement the standard Bartlett (BT) or high resolution Capon Maximum-Likelihood (ML) F-K estimation algorithms. As mentioned above the output from these programs may be thought of as a spatial Fourier transform estimated from an unevenly sampled grid. And as such the output beam response will be a convolution of the true wavefield spectra and the array factor (which includes applied filters). The BT and ML methods make no assumptions about the source or propagation media characteristics (other than the assumption that the wave fronts observed by the array are planar on the scale of the array diameter). Also, both routines are capable of producing a wide-band frequency domain array response via integration over a band of user-specified frequencies.

### **2. HARDWARE AND SOFTWARE REQUIREMENTS**

This section lists the system requirements for the operation of ESTK1D or ESTK2D in either a generic mode (machine independent) or for an 80386 PC workstation using executable files.

#### **2.1. Hardware requirements for use of executable files**

- a. 80386 IBM-compatible PC.
- b. 80387 Intel numeric processor.
- c. 1 MB of 32-bit extended RAM (approximate).
- d. VGA display monitor.
- e. 2-MB hard disk space (approximate).

## 2.2. Software requirements for use of executable files

- a. DOS 3.0 or higher.
- b. ASCII text editor.

## 2.3. Hardware requirements for creation of executable program based on generic source code

- a. 1.5-MB RAM (approximate).
- b. 2-MB hard disk space (approximate).

## 2.4. Software requirements for creation of executable program based on generic source code

- a. Fortran 77 compiler.
- b. ASCII editor.
- c. Object code linker.

## 3. PROGRAM DESCRIPTION

The central idea is to determine the wavenumber vector of a plane wave source by processing digital records of time domain waveforms from an array of passive sensors (geophones, microphones, hydrophones, radar antennas...) located at spatial coordinates separated by approximately one-half wavelength. Such a data set will be referred to as a signal vector. The wavenumber ( $\vec{k}$ ) of a propagating wave is a vector that defines the direction of wave propagation and the number of wave cycles per meter. It is often referred to as the spatial frequency of a wavefield. The wavenumber vector for a plane wave propagating with a vector velocity  $\vec{c}$  and frequency  $f$  may be written as

$$\vec{k} = \frac{2\pi f}{\vec{c}}. \quad (1)$$

In the  $xy$  plane such a vector will have components of

$$\vec{k} = \vec{k}_x + \vec{k}_y. \quad (2)$$

The program ESTK2D produces an estimate of the two-dimensional wavenumber ( $\vec{k}_x, \vec{k}_y$ ) by finding the vector wavenumber coordinates of the largest amplitude response of a beamformer. The output beam response is a three-dimensional regular grid space composed of  $k_x, k_y$  coordinate space versus power in decibels (for some specified frequency or band of frequencies). Using the vector coordinates of the peak power response in the output and a knowledge of the observed signal vector amplitude spectra, one can estimate the wave velocity and source bearing. In general the "ideal" spectral response will be best approximated by processing signals acquired from densely populated, geometrically large arrays, with long-duration time domain records, with high Signal-to-Noise Ratios (SNR), and a single plane wave stationary source.

Since ESTK1D and ESTK2D implement essentially the same beamforming algorithms, there is a high degree of commonality that will be addressed first. Individual descriptions of the unique characteristics of ESTK1D and ESTK2D follow in subsections 3.1 and 3.2. There is also extensive documentation present in the source code of each program. Only qualitative descriptions of the algorithms' mathematical foundations will be given. Unless otherwise stated, the reader should refer to Capon (1969) for mathematical detail.

The basic flow sequence for ESTK1D and ESTK2D maybe summarized by the following stages:

- a. The user supplies the program with a command file that specifies such things as the file name containing the signal vector, file name of output file, signal duration, array geometry and various processing options (see section 4 for details).

- b. The program opens and reads the file containing the signal vector for the array of sensors described in the command file (see section 4 for details).

- c. Each component of the signal vector (i.e., time series for a single sensor) is normalized by the sum of the absolute value of the largest positive and largest negative time domain peaks found in that signal. This ensures that errors attributable to small amplitude anomalies associated with the recording process are reduced by making all array elements of the signal vector the same amplitude.

- d. The temporal frequency spectrum of the SNR is estimated by utilizing the cross spectrum between two elements of the signal vector to obtain the magnitude squared coherence function, which in turn is used to form the SNR between the two signal vector components (Stearns and David 1988). The average for the array is formed by obtaining a sum of SNR estimates for all possible non-redundant pair-wise combinations of signal vector components and then normalizing by the number of combinations. An estimate of SNR is important for interpretation of the beamformer response function. Low SNR can severely reduce bearing (wavenumber) resolution and reliability.

- e. A block averaging FFT is then performed on each signal vector component. A time domain tapering function may be applied to each block of time series (see section 4). The block averaging process increases the stability of the estimated correlation matrix, which is needed to avoid singularities during matrix inversion when forming the ML beam response. Few singularities will be encountered if the number of blocks ( $M$ ) in each time series is greater than or equal to the total number of sensors ( $N$ ) in the array (see step h for further details on stabilizing the correlation matrix). The block average FFT processing also increases the reliability of the spectral estimate by reducing the variance. This is ac-

complicated at the expense of reduced resolution and increased bias.

f. Next, the program enters frequency and wavenumber/direction of look loops (repeating as necessary for steps g–k). The order of loop entry and wavenumber dimensionality depend on the specific program being run (see descriptions of unique characteristics in subsections 3.1 and 3.2).

g. The program estimates the correlation matrix from the block average spectra of the signal vector. This is a complex matrix with  $N$  order ( $N$  is number of sensors in the array) and  $M$  rank ( $M$  is number of blocks in the block average estimate of the signal vector estimate). For the correlation matrix to be non-singular,  $M \geq N$  (see Capon [1969] for mathematical details). If the user has specified the BT beamforming option, this is not a problem since the correlation matrix is not inverted.

h. If the program is forming the ML beam response, then it will invert the correlation matrix. There are many circumstances in which  $M < N$  and the correlation matrix will be singular. For example, a short duration time recording with 512 points and a time domain sampling frequency of 2000 Hz with 4-block FFT averaged spectra will have a frequency resolution of 15.6 Hz and contain only 64 points. Any array with a sensor population greater than four elements will produce a singular correlation matrix. To ensure that the matrix is non-singular, a small incoherent noise term ( $\lambda$ ) is applied to the correlation matrix (following Capon 1969). This guarantees the positive definiteness of the modified

correlation matrix and thus ensures non-singularity. The only practical effect the  $\lambda$  term has on the beam response is a slight reduction in the dynamic range of the beam response if one chose a large value of  $\lambda$ .

i. Note that the program will detect singular conditions during the inversion process and execution will be halted. The user will also be supplied with diagnostic information detailing the error condition.

j. The program forms a beam response at the current frequency and wavenumber (see Capon 1969 for details of calculation). Appropriate loops over frequency and wavenumber are continued.

k. If the beam response is formed over a band of frequencies, then the program forms a wide band beam response by integrating over the frequency band of interest. In most circumstances a narrow band beam response will produce the best result. However, in circumstances where the signal vector contains an impulsive (wide band) source with low SNR or there are multiple sources with differing narrow band spectra, then a wide band array response may provide the best results. In general, a wide band beam response will be lower in resolving capability.

l. The program writes processing results to the output file. Section 6 details the form and content of the output file.

### 3.1 Description of ESTK1D.FOR

The specific program flow of ESTK1D is shown in Figure 1. The program uses a one-dimensional wave-

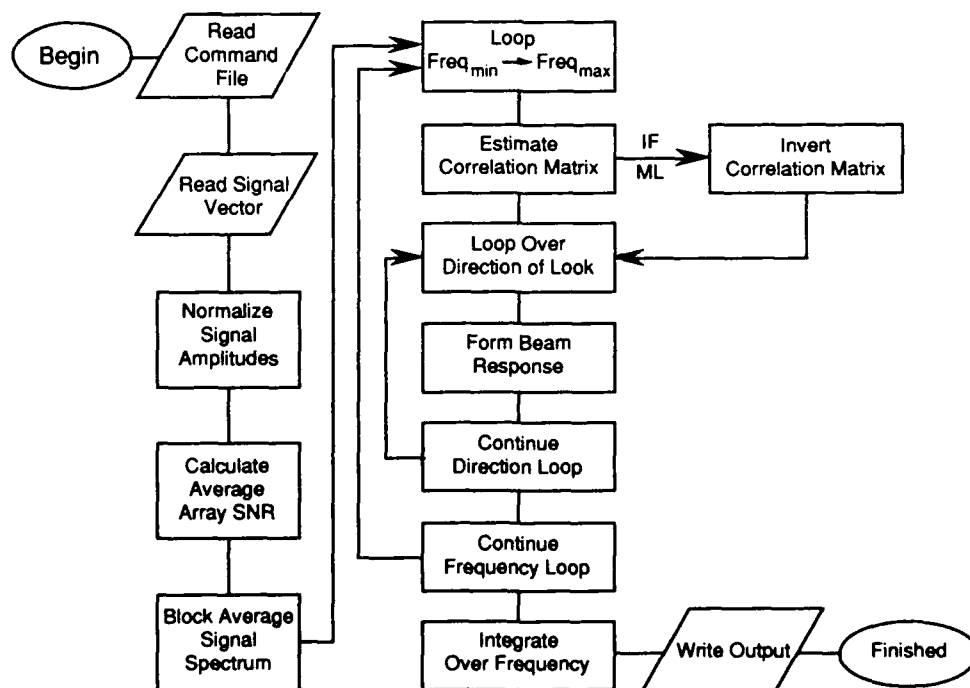


Figure 1. Flow diagram of ESTK1D.



number vector whose magnitude is defined by the user via the specification of the velocity of the propagating signal and the processing frequency. This assumes that the user has some previous knowledge of the velocity of the propagating signal. The wavenumber vector is then rotated in wavenumber space through an arc ranging from  $\pi/2$  to  $-\pi/2$  in incremental steps. Each step is said to be a "direction of look" ( $\theta$ ) for the array. Beamforming responses will peak when  $\theta$  corresponds to the bearing of the source. The output is expressed in  $\theta$  vs amplitude coordinate pairs.

This processing method is particularly suited when observation arrays are linear (one-dimensional) since any multidimensional wavenumber processing (such as that performed in ESTK2D) will not be able to resolve wavenumber components perpendicular to the axis of the array. The two-dimensional  $\theta$  vs amplitude output is also much easier to interpret. It should be noted that ESTK1D does not prohibit the use of two-dimensional observation arrays.

### 3.2 Description of ESTK2D.FOR

Program flow for ESTK2D is shown in Figure 2.

ESTK2D forms a beam response over a regularly spaced symmetrical grid in  $k_x, k_y$  vector space over a user-specified range of frequencies. This program is best suited for two-dimensional arrays. Source wavenumbers are indicated by the  $k_x, k_y$  coordinates of high amplitude zones. For example, a narrow-band 20-Hz source located at  $+20^\circ$  (counterclockwise relative to the  $+x$  axis) with a wave velocity of 200 m/s will be observed to have a wavenumber with components of  $k_x = -0.591 \text{ m}^{-1}$  and  $k_y = -0.215 \text{ m}^{-1}$ . The negative coordinate location in wavenumber space is ascribable to the array "seeing" an incoming wave front from the first quadrant of  $xy$  space (i.e., a negative wavenumber vector). To get the bearing of the source, one simply adds  $180^\circ$  to the angle defined by the wavenumber components. As mentioned in the preceding section, this processing method is not suited to linear arrays since the orthogonal direction to the array axis will not be resolvable. It does have the advantage of not needing prior knowledge of the velocity of the propagating wave. However, one should limit the extent of the wavenumber grid space to reduce the likelihood of misinterpreting grating lobes as secondary sources or introducing aliasing effects. (A

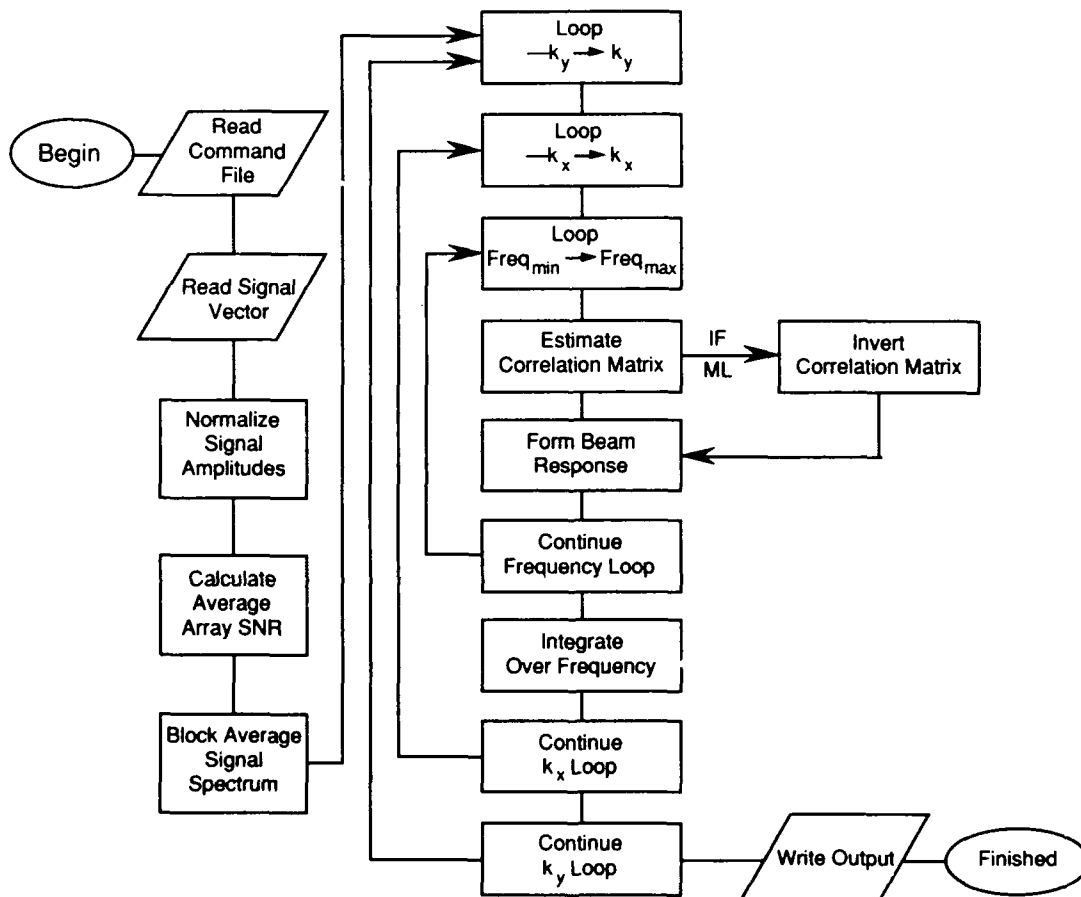


Figure 2. Flow diagram of ESTK2D.

grating lobe is a high amplitude zone that is associated with the periodicity of the beamformer response function [see Dudgeon and Mersereau 1984, pg 297].)

#### 4. COMMAND FILE OPTIONS AND STRUCTURE OF INPUT FILES

This section will detail the required structure and content of the various user supplied input files. Two files are required to operate ESTK1D and ESTK2D—a command file that controls the program operation and a data file that contains the time series (signal vector) observed by the array.

##### 4.1 Command file

To operate ESTK1D or ESTK2D, the user must supply the program with the name of a command file that contains all the processing options, array element locations and associated channel numbers, the name of the file that contains the time series data for the array (i.e., the signal vector), and the desired name of the output file to which the beam response information will be written. The name of the command file may be specified on the command line that invokes the program or at a program prompt. Any valid file name for the operating system that is 12 characters or fewer will be accepted by the program as long as the file has the required format. The command file may be created using any ASCII text

editor. In general the command file will contain two types of lines. The first type is a header line that is present only for legibility and is discarded by the program. Any alphanumeric ASCII characters are valid on this type of line. The second type of line is a control line that has two to three fields containing input parameters followed by a concise description field (for clarity). Valid field delimiters are blank spaces or commas. The presence of description fields on a control line is not required.

##### 4.1.1 ESTK1D command file

Figure 3 is an example of the command file needed to operate ESTK1D. Only information supplied inside the box is to be specified in the command file. The box and the line numbers outside the box are for reference only. In this example any header lines following line 21 will be ignored. Following is a line-by-line description of command file shown in Figure 3.

1) This is a header line that provides the user with file description information.

2) The first field contains the name of the data file. Any character string with 12 or fewer characters that is acceptable to the operating system is valid. The second field provides a brief descriptor. It has no impact on program execution and may be blank.

3) The first field contains the name of the output file to which the beamformer response and other processing

```

1) ===== ESTK1D SCRIPT FILE =====
2) DATFIL      FILE NAME CONTAINING WAVEFORM DATA
3) OUTFIL.M9   OUTPUT FILE FOR PROCESSED SPECTRAL DATA
4) 1           TYPE OF TIME DOMAIN TAPER APPLIED TO EACH BLOCK
5) 0           BEAM FORMING OPTION 1=BARTLETT, 0=MAX LIKELIHOOD
6) 1           TEST CORRL. MATRIX INV, (ONLY FOR ML, 1=YES, 0=NO )
7) 4           NUMBER OF CHANNELS READ
8) 1.E-5       MATRIX STABILIZATION FACTOR (LAMBDA)
9) 256, 2      # OF PTS IN TIME SERIES, # OF BLOCKS IN FFT
10) .0005      TIME INTERVAL BETWEEN SAMPLES ( IN SEC. )
11) 346.       WAVE VELOCITY (m/S) FOR PHASE OF INTEREST
12) 15.,15.    MIN FREQ (FMIN), MAX FREQ. (FMAX)
13) 1          ANGULAR RESOLUTION OF CALCULATION (IN DEGREES)
14) 1          TOGGLE FOR AMP VS. FREQ/THETA SURFACE, 1=YES 0=NO
15) 1          DEBUG TOGGLE: 1=ON, 0=OFF
16) ==ARRAY ELEMENT LOCATIONS BY CHANNEL NUMBER ==
17) CHANNEL #   X           Y
18)      1      0.0        0.0
19)      6      0.0        7.0
20)      8      7.0        0.0
21)      4      7.0        7.0

```

Figure 3. Example command file needed to operate ESTK1D.

information will be written. Any character string with 12 or fewer characters that is acceptable to the operating system is valid. The second field provides a brief user description.

4) The first field specifies the time domain taper that will be used in the block average FFT and in the estimation of the average array SNR. Valid numbers are integers from 1 to 6. The following lists the valid parameters and their associated window:

- |                              |               |
|------------------------------|---------------|
| 1 = Box car (no taper)       | 4 = Hanning   |
| 2 = 10% cos taper on box car | 5 = Hamming   |
| 3 = Triangular               | 6 = Blackman. |

In general, the highest spectral resolution is obtained with the box car window and the lowest resolution is given with the Blackman filter. However, there is the well known trade-off between resolution and side lobe decay, the box car window having the slowest side lobe decay and the Blackman window giving the fastest side lobe decay rate. See Stearns and David (1988, p. 165) for the details of the effects that the various windows have on spectral estimation. The second field in line 4 provides a brief description.

5) On line 5 the first field defines the beamforming method to be applied. An integer value of 0 or 1 is valid input—0 specifies that the high resolution Maximum Likelihood (ML) method (Capon 1969) will be performed, while 1 means that the standard Bartlett (BT) beamformer will be implemented. The second field is a description area.

6) The first field allows the option of testing the quality of the correlation matrix inversion if the ML method is chosen on line 5. If the BT method is specified, then line 6 has no effect on the program flow. Acceptable values for this field are 0 or 1. If 0 is chosen then no inversion test will be performed. If 1 is chosen here and if ML beamforming was specified on line 5, then the inversion will be tested by forming the identify matrix from the inverted correlation matrix and the original correlation matrix. Diagonal elements of the resulting matrix are compared to 1 and off-diagonal elements are compared to 0. Significant deviations ( $\pm 1 \times 10^{-6}$ ) from the identify matrix are reported to the user and an option of aborting the program is made available. If the user decides to continue, a warning is reported in the output file and to the terminal at completion of the program execution. The second field is a descriptor area.

7) The first field specifies the total number of sensor elements or channels contained in the data file given on line 2. Acceptable values range from 1 to 24. A minimum of two channels is required to form a beam. It is strongly recommended that three or more channels be used. The second field is a descriptor area.

8) On line 8 the first field defines the value of  $\lambda$  used to stabilize the correlation matrix for inversion if the

ML option is chosen on line 5. It has no effect if the BT method is chosen. In general  $\lambda$  can be on the order of  $10^{-5}$  smaller than the magnitude of the diagonal elements of the correlation matrix. The user should experiment to find the smallest value for  $\lambda$  that will produce a non-singular correlation matrix. If the number of blocks in the FFT (specified in field 2, line 9) is larger than the number of sensor elements (given in field 1, line 7), then  $\lambda$  may be 0. The second field is a descriptor area.

9) There are two program control fields in line 9. The first field specifies the number of points in the time series of each channel of the signal vector file. This number must be a power of 2 and less than or equal to 1024 points. The second field specifies the number of non-overlapping blocks to be used in the block average FFT performed on each component of the signal vector. It must also be a power of 2. The third field is a descriptor area.

10) The first field specifies the time sample interval between each time series datum. The second field is a descriptor area.

11) The first field on this line defines the velocity of the propagating signal. Errors of  $\pm 10\%$  should not dramatically affect the beam response for a well designed array. To get the best result in an ambiguous velocity setting, the user should make several runs with a range of velocities and pick the result with the largest beam power (reported in the output file). The second field is a descriptor area.

12) There are two program control fields in line 12. These two fields define the upper and lower frequencies to be processed. If the first and second fields contain the same value or differ less than the frequency domain sample interval, then only the first frequency will be processed. The resulting beam response is said to be narrow band. If the frequency specified in the second field is greater than the first by more than  $N$  times the spectral resolution after block averaging has been done, then an array response will be formed for each of the  $N$  frequencies in the band and an integration will be done to produce a wide band array response to the beamformer. The value of the second field must be equal to or larger than the value of the first field. If running the MS-DOS executable files, then setting both fields to 0 will result in a graphical display of the average spectra for the entire signal vector. (Note: in the generic source code listing no spectral display is given and 0 in these fields is not valid.) The user will then be prompted to specify the processing band of interest. This option is convenient if one is dealing with unfamiliar signals. To obtain an acceptable beam response, the choice of frequency processing bounds should be confined to that portion of the signal spectra with the largest amplitudes.

In most cases the user will find that a narrow band processing strategy provides the best results. The third field is a descriptor area.

13) The first field specifies the resolution of the incremental angular steps through which the observation wavenumber vector will be rotated. This value is given in degrees and must be greater than or equal to 1. Note that output will be reported in  $\theta$ , beam power coordinate pairs ranging from  $90^\circ$  (+y) to  $-90^\circ$  (-y) in the increment defined by this field. The second field is a descriptor area.

14) The first field defines the state of a toggle. Acceptable values are 0 or 1. A value of 1 will direct the program to output the narrow band beam response for each frequency processed in addition to the integrated wide band response. This allows one to make a more detailed analysis of the beam response. A value of 0 is the off state and only the integrated response will be reported (assuming a wide band response was requested by the user). Note that in the on state, if a large number of frequencies are processed, then the output file may become very large. The second field is a descriptor area.

15) The first field defines the state of a debugging toggle. Acceptable values are 0 or 1. A value of 1 is the on state that directs the program to open a file named "debug9e" to which a very detailed report of intermediate program calculations will be written. This is a hard way of monitoring the program behavior when troubleshooting the source code. It is recommended that this

toggle stay in the 0 state unless the Fortran source code is being modified. The second field is a descriptor area.

16) Header line used for clarity. Any string of alphanumeric characters is valid on this line.

17) This header line is used for clarity. Any string of alphanumeric characters is valid on this line. In this case the header line provides column titles for the array channel number and xy coordinate locations.

18-21) These four lines associate a channel number, which is the identifier in the signal vector file (see subsection 4.1.2), with an xy coordinate position. The number of lines in this section of the command file must be the same number specified in the first field of line 7 given above. There are three fields in these lines. The first field contains the channel-sensor number, the second and third fields contain the x-axis and y-axis coordinates of the sensor's spatial location in the array. Error in the location of these channel coordinates should be *much* smaller than  $1/4$  of the shortest wavelength expected to be observed by the array. The channel-position lines do not have to be in any specific order and the coordinate system origin need not be at an array element location.

Any line following the last channel-position line will be ignored.

#### 4.1.2 ESTK2D command file

Figure 4 is an example of the command file needed to operate ESTK2D. This command file shares many of

```

1) ===== ESTK2D SCRIPT FILE =====
2) DATFIL      FILE NAME CONTAINING WAVEFORM DATA
3) OUTFIL.M10  OUTPUT FILE FOR PROCESSED SPECTRAL DATA
4) 1           TIME DOMAIN TAPER APPLIED TO EACH BLOCK
5) 0           BEAM FORMING OPTION 1=BARTLETT, 0=MAX LIKELIHOOD
6) 1           TEST CORRL. MATRIX INV, (ONLY FOR ML, 1=YES, 0=NO )
7) 4           NUMBER OF CHANNELS TO BE READ
8) 1.E-5       MATRIX STABILIZATION FACTOR (LAMBDA)
9) 256, 2      # OF PTS IN TIME SERIES, # OF BLKS IN FFT
10) .0005      TIME INTERVAL BETWEEN SAMPLES ( IN SEC.S )
11) 15.,15.    FMIN, FMAX
12) .8,.8      KX_MAX, KY_MAX, RANGE WILL BE -KX_MAX TO +KX_MAX
13) 30, 30     NUMKX, NUMKY : GRID WILL BE, 2(NUMKX+1) BY 2(NUMKY+1)
14) 0          DEBUG PARAMETER 1=ON, 0=OFF
15) ==ARRAY ELEMENT LOCATIONS BY CHANNEL NUMBER ==
16) CHANNEL #   X           Y
17)      1      0.0         0.0
18)      6      0.0         7.0
19)      8      7.0         0.0
20)      4      7.0         7.0

```

Figure 4. Example command file needed to operate ESTK2D.

12) This is a three-field line. The first field defines the magnitude of the largest  $k_x$  abscissa value (KX\_MAX) in vector wavenumber space. The second field defines the magnitude of the largest  $k_y$  abscissa value (KY\_MAX) in vector wavenumber space. Thus, the beam response function will be found for a regular grid space that ranges from -KX\_MAX to +KX\_MAX and -KY\_MAX to +KY\_MAX. These values should be chosen carefully so as to avoid evaluating the beam response function in a region of wavenumber space that will be dominated by large sidelobe effects (i.e., avoid evaluating wavenumbers beyond the spatial Nyquist wavenumber, which is defined by the smallest interelement array spacings in the  $x$  and  $y$  directions). Any positive real number is a valid entry for these two fields. The third field is a description area.

14) In ESTK2D the debug toggle is non-functional. Later versions will incorporate this feature. In this version it is a header line.

ESTK1D and ESTK2D use the same data file format. The structure of the data file is fairly simple. There are three required elements:

- c. After the channel line follows a series of lines containing the observed time series for that channel. The number of points in each time series field must be a power of 2. Each line of each time series must contain at least one real number and may contain as many as five real numbers. In cases where there are multiple time

The program reads each component of the signal vector in the data by searching for the line of Es. Once found, the channel number line is read and stored in an array. Then the time series is read and stored in the signal vector array. The process is repeated for the next component of the signal vector until all components have been read. Searching for the line of E's allows for a diversity of data file formats and also allows a large description header for each component of the signal vector.

The installation and operation of ESTK1D and ESTK2D are detailed below. This description is based on the code implemented for 80386 microcomputers using the MS DOS version 3.0 (or higher) operating system. The generic versions (assumed to be UNIX implementations) of these programs follow closely the sequences presented below. Deviations are noted in parentheses. Required terminal input is in **bold print**.

c. Read the “read.me” file for the latest program information, manual updates and miscellaneous details.

There are two ways of initiating the execution of these programs. Method 2 is preferred because it requires less program interruption and is well suited for autonomous operation via batch file execution (or a script file in a UNIX environment).

The user will immediately be presented with a program identification statement followed by a request for terminal input of the command file name (see section 4.1). After the terminal provides the name of the command

file, a program status display reporting the progress made in reading the command file will be given.

### 5.2.2 Method 2

At the DOS prompt, type the root of the program name, a space, the name of the command file, then a carriage return. For example:

```
c:>ESTK2D SCRIPT.10A.
```

The user will immediately be presented with a program identification statement followed by a program status display reporting the progress made in reading the command file "SCRIPT.10A."

On a UNIX platform, the equivalent invocation method may be issued with a command line such as:

```
%estk2d < script.10a .
```

### 5.3 Program status reports and required terminal input

The particulars of program flow depend upon the command file options chosen and the signal vector characteristics. In this discussion it will be assumed that there are no errors in the format of either the command file or the data file containing the signal vector (see subsection 5.4 for error conditions). The terminal will display a status report as each major calculation or input-output operation is completed. If there are no problems with the command or data files, then the only response required from the terminal will be in cases where the minimum and maximum frequency band limits given in the command file are equal to zero. Such a condition is only valid for 80386 implementations of these programs. If no frequency bounds are given, the user will be informed that a graphical display is being formed of the block average signal vector spectra for the entire observation array. Once the program is ready to display the result, a prompt is given requesting the user to strike any key. The spectra are then plotted on the terminal display.

The user should note the range of frequencies over which the source has a significant amplitude. Another carriage return will erase the display and a prompt will appear asking if the user wants to see some subset band of frequencies. Valid responses to this prompt are 1 or 0—an affirmative response being 1 and 0 being negative. If the user answers in the affirmative, a prompt will appear asking for the values of the upper and lower bounds of the band of interest and a graphical display of that interval will be shown in the same manner as given above. After the second display, or if the user responded negatively to the first prompt, the user will be prompted for the band of frequencies to be processed. Valid re-

sponses are any positive, real, non-zero numbers where the second number is larger than or equal to the first. These frequency values should be at or near the spectral peaks of the signal vector. No further terminal input should be required unless an error condition is encountered with the signal vector. It should be noted that the user may terminate program execution at any prompt that requests terminal input by entering a key combination of **CONTROL+C** or **CONTROL+BREAK**.

### 5.4 Possible error conditions

There are several error conditions and accuracy checks that may require a terminal response or cause the program to abort. These stages in the program flow are listed and explained in the order in which they may be encountered. There are only minor differences between ESTK1D and ESTK2D and there are no differences between the 80386 and generic versions of the programs.

a. A fatal error (causes program execution to abort) will occur in the calculation of the array SNR if an invalid number was specified for the time domain window function in the command file (line 4, field 1, in both ESTK1D and ESTK2D command files). A message indicating the problem and the location of its occurrence is displayed on the terminal at the time of program suspension.

b. A fatal error will occur during the inversion of the spectral correlation matrix (when forming the maximum likelihood beam response) if that matrix is found to be numerically singular. Matrix singularity is determined by calculating a condition number (Dongarra et al. 1979). If the condition number is numerically 0 (i.e., very small), then the matrix is effectively singular. A message stating the matrix singularity condition and the value of the condition number is displayed at the time of program suspension. To correct the problem try making  $\lambda$  larger (specified on line 8, field 1, in the command files of both ESTK1D and ESTK2D).

c. If the command file (field 1, line 6) specified that the quality of the correlation matrix inversion should be tested, then the identity matrix is computed using the original correlation matrix and the inverted matrix. The real part of the diagonal elements of the resulting matrix are compared to 1.0 with a tolerance of  $\pm 1 \times 10^{-5}$  and the magnitude of the imaginary part is compared to 0.0 with a tolerance of  $1.0 \times 10^{-7}$ . The magnitudes of off-diagonal elements are compared to 0.0 with a tolerance of  $1.0 \times 10^{-7}$ . If any element fails to pass the appropriate test, then program flow is halted and a message is displayed stating the error condition. The user is provided the option of continuing or aborting the program. If the user opts for continued execution then a warning message will be written in the output file indicating that the

quality of the matrix inversion may have been poor. To correct the problem try making  $\lambda$  larger (specified on line 8, field 1, in the command files of both ESTK1D and ESTK2D).

d. Another quality check is done on the beam response for both the BT and ML at every observation wavenumber vector. The beam response is a calculation that estimates the power of the incident energy at each observation vector. It is formed from a complex steering vector and the complex correlation matrix. A power function is a positive, real-valued function. Thus, the magnitude of the imaginary component of the beam power function should be numerically 0 compared to the real part and the real part must be greater than 0. These properties are checked at every evaluation point. If either the real or imaginary parts of the calculated beam response is found to be inconsistent with the properties of a power function, then program flow is halted, and an error condition message is displayed. The user then has the option of continuing or aborting the program. If the user opts for continued operation, then a warning message will be written to the output file.

## 6. DESCRIPTION AND EXPLANATION OF OUTPUT FILE FORMAT

The output files from ESTK1D and ESTK2D contain a variety of processing and signal characteristic information in addition to the beam response power function. There are header lines in the output that describe each item or section of information. A brief sequential description of each major section of the output files is given here.

a. The first few header lines provide the names of the command file, data file, the program that produced the output file, the date the file was made and the name of the output file.

b. The next major section of header lines provides a record of the processing options chosen in the command file as well as information on the block averaging process performed on the signal vector.

c. The third major section provides three important signal characteristics. The first is the normalization factor applied to each channel of the signal vector. These values should not differ by more than  $\pm 5$ –10%. The second item of information on the signal is the spectrum of the average array SNR. The SNR is a major factor affecting the resolution of the beamformer. In general the processing bounds should be chosen in the vicinity of a local SNR maximum. It should be noted that the estimation of SNR is very sensitive to the number of blocks and the time domain window taper used in the block average FFT process. There may also be very

large SNR associated with frequencies that have very little signal or noise energy (ratios of very small numbers can often produce large values). The output file also provides a normalized average array amplitude spectrum of the signal vector. The processed frequencies in the beam response should be in the vicinity of the spectral maxima. Energy from frequencies other than the frequency being processed will be seen as noise energy leaking into the beam response. If one is forming a beam at a portion of the spectrum with very little energy, then the beam response may be significantly degraded in resolution and accuracy.

d. The next section provides a synopsis of the beam response maxima at each frequency processed.

e. The final set of lines are discrete sample points of the beam response function. In ESTK1D these lines contain bearing versus beam power (expressed in decibels). In the case of ESTK2D this section contains the beam response as a function of two-dimensional wavenumber space. Thus, the output is expressed in terms of  $k_x$ ,  $k_y$  and beam power (in decibels).

## 7. PROGRAM ASSUMPTIONS AND PERFORMANCE

There are several assumptions made in the mathematical development of these array processing beamformers that the user should keep in mind, particularly when applying the high resolution ML method. The first assumption is that all processed signals are effectively plane waves on the scale of the array aperture. This assumption is required because the beamformer looks for linear phase shifts in the spectra between components of the signal vector. Any signals that significantly deviate from this rule will be seen as high energy incoherent noise by the beamformer, resulting in a very poorly focused or an unfocused beam. The second key assumption is that the noise field observed by the array has an uncorrelated Gaussian distribution. All undesirable correlated energy should, if possible, be filtered before beamforming is attempted. Another key requirement is that the source should be stationary over the interval of recording or processing.

Typical execution times on a 25-MHz 80386 microcomputer, when processing small data sets and a narrow band beam response such as the Appendix A examples, is on the order of 30 seconds for ESTK1D and 90 seconds for ESTK2D. Data sets with large arrays and long time series records may take several minutes to process. One of the key command file parameters affecting the execution time in ESTK2D is the number of grid points at which the beam response must be calculated (see the description of line 13 in subsection 4.1.2). Increasing

the frequency bandwidth over which the beam response is found will also dramatically increase the processing times for both ESTK1D and ESTK2D.

## 8. LITERATURE CITED

**Capon, J.** (1969) High-resolution frequency-wave-number spectrum analysis. *Proceedings of the IEEE*, **57**: 1408-1418.

**Capon, J., R.J. Greenfield and R.J. Kolker** (1967) Multidimensional maximum likelihood processing of a large-aperture seismic array. *Proceedings of the IEEE*, **55**(2): 192-211.

**Dongarra, J.J., C.B. Moler, J.R. Bunch and G.W.**

**Stewart** (1979) *Linpack Users' Guide*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics.

**Dudgeon, D.E. and R.M. Mersereau** (1984) *Multidimensional Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., p. 400.

**Hahn, W.R.** (1975) Optimum signal processing for passive sonar range and bearing estimation. *Journal of the Acoustic Society of America*, **58**: 201-207.

**Johnson, D.J.** (1982) The application of spectral estimation methods to bearing estimation problems. *Proceedings of the IEEE*, **70**(9): 1018-1028.

**Stearns S.D. and R.A. David** (1988) *Signal Processing Algorithms*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., p. 349, Appendix A.



## APPENDIX A: EXAMPLES OF PROGRAM OUTPUT

A graphical display of an example data set and the processed output files from ESTK1D and ESTK2D are given below. The command files used are listed in subsections 4.1.1 and 4.1.2.

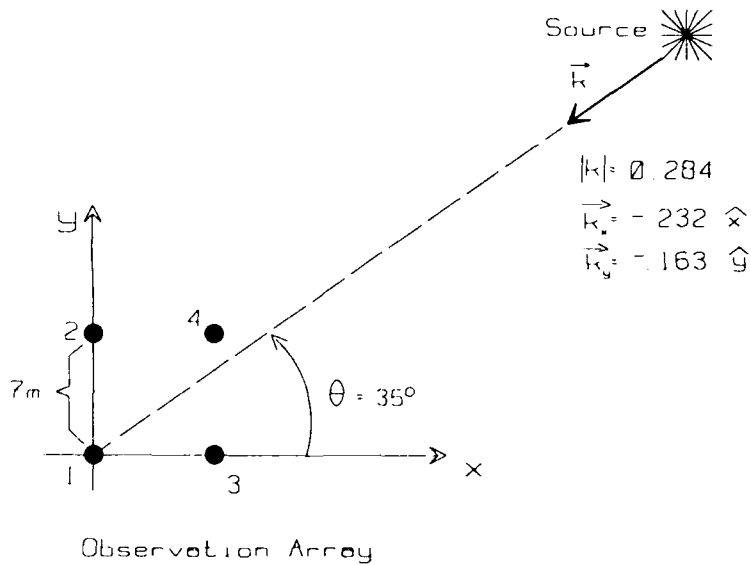


Figure A1. Example observation array and source parameters. Source propagation velocity is 346 m/s with a center frequency of 15.6 Hz.

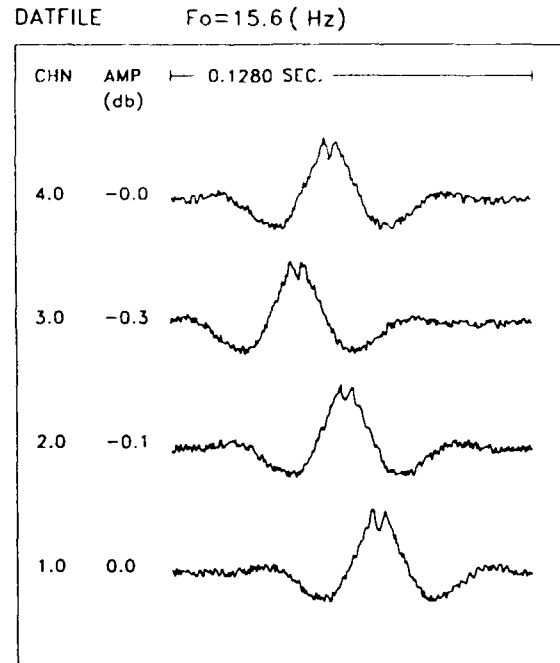


Figure A2. Example signal vector waveforms.

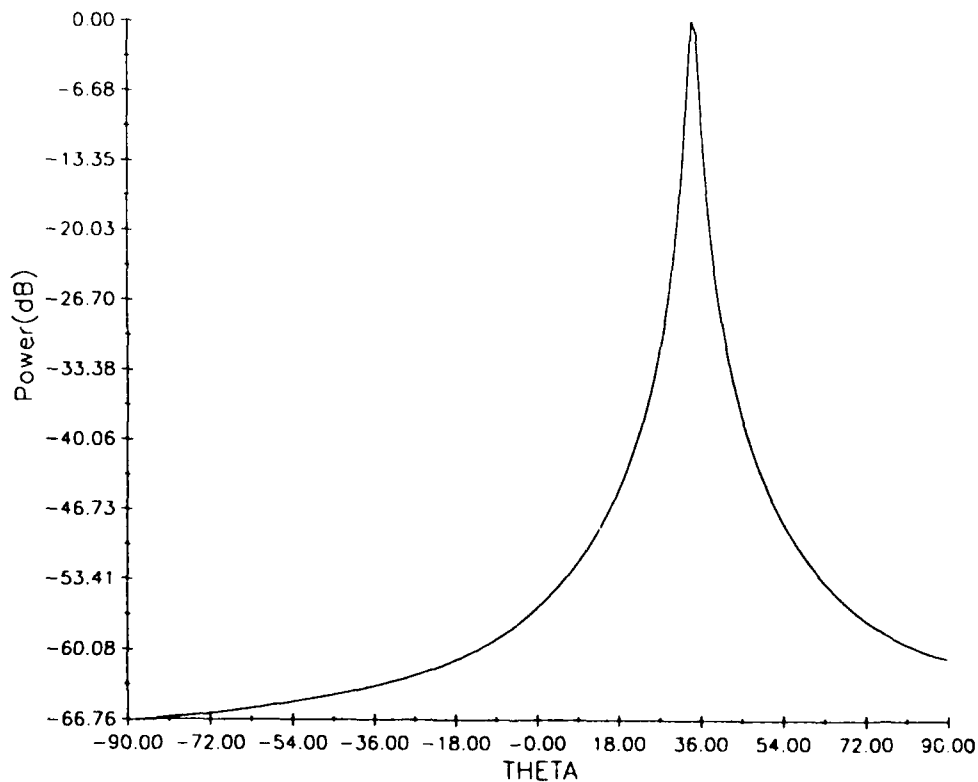


Figure A3. ML beam response produced by ESTK1D.

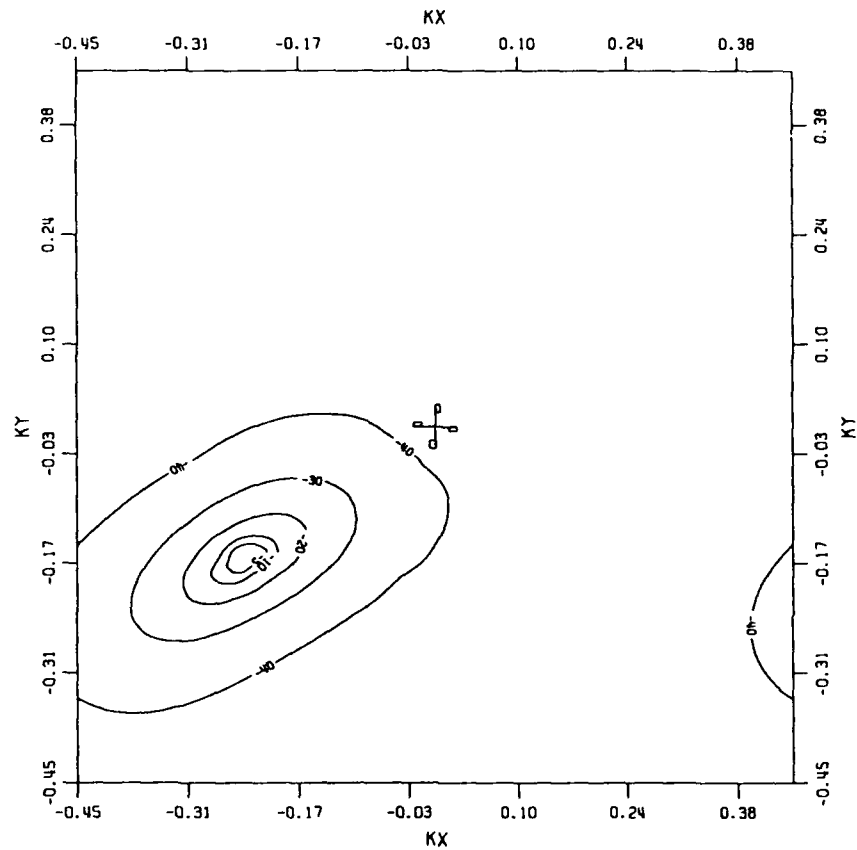


Figure A4. ML beam response in two-dimensional wavenumber space produced by ESTK2D. Peak power response was found to be at  $k_x = -0.24$ ,  $k_y = -0.165$ . This gives a source bearing of  $34.5^\circ$  with a propagation velocity of 337 m/s. The smallest contour shown is -5 dB.

## APPENDIX B: LIST OF REQUIRED FILES

As mentioned in the introduction, the source code and executable files are available from the author on a floppy disk or via an electronic file transfer using FTP. Listed below are the various files needed to generate an executable version of ESTK1D and ESTK2D.

### B1. Files required to run ESTK1D

ESTK1D.FOR	Main program.
PACK9E.FOR	Auxiliary subroutines.
ICOR9E.FOR	Correlation matrix inversion subroutines.
INTEGRAL.FOR	Integration subroutines.
ESTK1D.INC	Include file contains array dimensioning parameters and program documentation notes.
SCRIPT9E	Command file, may be any 12-element string of alphanumeric characters (see example command file in subsection 4.1.1).

### B2. Files required to run ESTK2D

ESTK2D.FOR	Main program.
PACK10A.FOR	Auxiliary subroutines.
ICOR10A.FOR	Correlation matrix inversion subroutines.
INTEGRAL.FOR	Integration subroutines.
ESTK2D.INC	Include file contains array-dimensioning parameters and program documentation notes.
SCRIPT.10A	Command file, may be any 12-element string of alphanumeric characters (see example command file in subsection 4.1.2).

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestion for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1991		3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE  User's Manual for ESTK1D.FOR and ESTK2D.FOR Wavenumber Estimation Routines				5. FUNDING NUMBERS  PR: 4A762784AT42 TA: FS WU: 017	
6. AUTHORS  Mark L. Moran					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  U.S. Army Cold Regions Research and Engineering Laboratory 72 Lyme Road Hanover, New Hampshire 03755-1290				8. PERFORMING ORGANIZATION REPORT NUMBER  Special Report 91-15	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Office of the Chief of Engineers Washington, D.C. 20314-1000				10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.  Available from NTIS, Springfield, Virginia 22161.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  This document describes the operation and structure of the Fortran programs ESTK1D.FOR and ESTK2D.FOR. These frequency domain wavenumber estimation programs implement either the Bartlett or the high-resolution Capon (1969) maximum-likelihood beamformers. The program ESTK1D.FOR forms a beam response based on a one-dimensional observation wavenumber that is rotated through wavenumber space to determine the spatial bearing of a plane wave source. ESTK2D.FOR uses the beam response in two-dimensional wavenumber space to estimate the two-dimensional wavenumber vector of a plane wave source. The discussion presented in this paper focuses on the operational details of installing and running ESTK1D and ESTK2D.					
14. SUBJECT TERMS Array processing Beamforming Cold regions  FORTRAN programs Geophones Geophysics  User's Manual Wavenumber estimation				15. NUMBER OF PAGES 21	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT		